View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Electronic archive of Ternopil National Ivan Puluj...

повідомлень;

- Виклик застосунків на Rust, як консольних утиліт;
- Взаємодія з допомогою FFI (Foreign functional interface).

Ось приклад написання модуля за допомогою FFI;

- Опис типу скомпільованого файла та залежностей проекту на Rust

```
# Cargo.toml[package]name = "test_addon"version = "0.0.1"authors =
    ["Ivan Lavriv <lavriv92gmail.com>"][lib]name =
    "simple_addon"crate_type = ["dylib"][dependencies]# List of your
    dependensies mongodb = "*"
```

- Реалізація модуля мовою Rust

```
// lib.rs #[no_mangle]fn public_function() {    // do something}
```

- Реалізація модуля обгортки (в цьому випадку мовою python).

```
# module.py import ctypesmodule =
    ctypes.cdll.LoadLibrary('./path_to_compiled_addon')mudule.public_
    method()
```

У разі реалізації розширення за допомогою FFI існує типова проблема синхронізації типів даних та семантичних особливостей між двома мовами. Для цього існують вже готові бібліотеки, які реалізовують особливості інших мов з допомогою особливостей Rust (rust-cpython, rust-rmi, тощо). Ось приклад використання rust-cpython

```
use std::any::Any;extern crate cpython;use
    cpython::ObjectProtocol;use cpython::{PyList, PyDict, PyObject,
    Python};#[no_mangle]fn get_data_as_dict() -> PyDict
    {}#[no_mangle]fn get_data_as_list(iterable: &[f64]) -> PyList
    {let python = Python(); let result = iterable.map(|&elem: &T|
    { match elem { Some(elem) => process(elem), None =>
    Python.None()}}); return PyList(python, result);}
```

Як висновок можна сказати, що Rust є сучасною, швидкою та безпечною мовою програмування, яка дає змогу програмісту цікаво і якісно вирішувати системні задачі в програмуванні.

Використання вільного програмного та апаратного забезпечення для точної реєстрації часових міток подій.

Мартинюк-Лотоцький К.П., Сергеев О.В.

1. Міжнародний центр Астрономічних і медико-екологічних досліджень, національна академія наук України; 2. Львівський національний університет імені Івана Франка, Астрономічна обсерваторія, langurek@gmail.com

Precise timing is one of the keystones for modern astronomical observations. The local timing service of the telescope Zeiss-600 at International Center for Astronomical, Medical and Ecological Research was upgraded. Its hardware consists of the GPS receiver and micro-PC RaspberryPi with RS422-to-TTL convertor. The corresponding software (Raspbian OS based) provides the necessary operations: timing data storage on SD card with access via ssh or ftp, providing local ntp server (it is synchronized over the public Internet and/or by PPS from GPS receiver). The developed ntp server maintains time to within 0.004 ms jitter

(accuracy), that is 100 times better than without GPS.

Для забезпечення точності і достовірності астрономічних спостережень необхідно забезпечити реєстрацію часу з високою точністю і стабільністю. Раніше використовували службу часу, на основі прийому сигналів точного часу по радіо. Її недоліки: спонтанне завмирання радіосигналу, під час зміни погоди сильні завади, тощо. З появою Інтернету почали використовувати ntp-сервери часу. Перевага: відразу можна отримати час з похибкою менше 20 мс, що є для більшості задач достатньо. Недолік: якщо немає зв'язку з Інтернетом, то час теж не можна синхронізувати.

На даний час існує достатньо багато високоточних систем реєстрації моментів подій на основі GPS-технологій. В цій роботі обговорюється використання мікрокомп'ютера RaspberryPi для узгодження інтерфейсів і організації роботи GPS і служби часу в цілому, включно з записом моментів подій, перетворенням форматів файлів і дистанційним доступом до них. Причому, служба часу повинна забезпечувати неперервний режим реєстрації моментів часу в складних умовах, коли сигнал від супутників системи GNSS (GPS) може тимчасово пропадати.

Для забезпечення вищої точності і надійності було встановлено GPS-приймач з входом маркера подій, який дає похибку реєстрації однієї події менше 100 нс. GPS має вихідний PPS-сигнал, по інтерфейсу RS-422 у форматі TSIP можна отримати дані про дату, час, точність реєстрації координат/часу, кількість супутників та інше.

GPS під’єднано до мікрокомп'ютера RaspberryPi через перехідник RS-422 -> RS232(TTL), сигнал PPS заведено на один з входів GPIO. На системі Raspbian (Debian Linux) запущено локальний ntp-сервер який синхронізується по мережі з ntp-серверами інтернету, а також уточнює час по PPS сигналу з GPS. Для реєстрації моментів відкриття/закриття затвору світлоприймального пристрою (ПЗЗ-камери, ФЕП тощо) зроблено перехідник (з опто-роз'вязкою) з вхідного струмового сигналу на сигнал event GPS (RS-422). Дані з GPS у форматі Trimble TSIP розшифровуються Perl-скриптом і далі зберігаються локально у файли і sql-бази.

Інформація про статус служби часу, про файли з моментами часу спрацьовування затвору відображаються на web-сторінці сервера, який запущено на легкій платформі lighttpd. Також дані можна завантажити по rcp(ssh), ftp, які періодично синхронізуються з резервним сервером по rsync, і до якого є доступ по samba для win-машин.

Система створена і введена в роботу на телескопі Zeiss-600 (пік Терскол, МЦ АМЕД) для реєстрації фактичних моментів відкриття/закриття затвору астрономічного світлоприйомного пристрою (CCD SBIG, Apogee).

Порівняльні характеристики роботи локального ntp-сервера приведено у таблиці. Значення зміщення (offset) і тремтіння (jitter) міток зовнішнього ntp-сервера (супутниковий інтернет) на декілька порядків більші ніж для міток на основі PPS від GPS.

Тип мітки часу	delay, ms	offset, ms	jitter, ms
ntp	0.617	21.203	0.486
PPS	0.000	-0.005	0.004

Джерела:

1. http://www.raspberrypi.org
2. http://www.satsignal.eu/ntp/Raspberry-Pi-quickstart.html

Месенджер для платформи Android на основі протоколу WebRTC

Муха Б.М., Шпак З.Я.

Національний університет «Львівська політехніка», muhabohdan@hotmail.com

The new open source protocol WebRTC for browser-to-browser real-time communication is analyzed. WebRTC-based application named Q-municate for Android platform is presented. Q-municate supports voice calling, video chats and group conferences, peer-to-peer data and file sharing without the need of either internal or external plugins. Designed messenger enforces usage of the encryption for both the media and the signalling.

Кожного дня люди передають один одному великі потоки інформації, спілкуються між собою за допомогою телефонів, чатів, відеоконференцій, обмінюються даними, фотографіями, файлами тощо. Передача відео і аудіо-інформації у режимі реального часу стала важливим етапом у розвитку нових Веб-технологій. Масового поширення набули засоби передачі медіа-інформації, що дають змогу здійснювати аудіо та відео-дзвінки, а також надсилати текстові повідомлення і медіаконтент. Актуальною і важливою є задача організації захищеного з'єднання між двома й більше учасниками розмови та безпосереднього передавання інформації до адресата без використання проміжних ланок.

Класичні протоколи передачі даних, такі як SIP та RTP, не задовольняють цих вимог, оскільки вони використовують проміжні сервери для забезпечення постійного з'єднання між клієнтами, що призводить до зниження захищеності. Протокол режиму реального часу RTP використовує динамічно призначувані адреси портів, чим створює труднощі у процесі проходження міжмережевих екранів. Для обходу цієї проблеми зазвичай використовують STUN-сервер. Це призводить до різкого зниження швидкості передачі даних та істотного спотворення відеозображення. Також недоліком є те, що STUN- і SIP-сервери, які необхідні для встановлення і підтримки з'єднання між клієнтами, не є безпечними ланками. Специфікація